

# Quantifying Information Leakage in Finite Order Deterministic Programs \*

Ji Zhu<sup>†</sup> and Mudhakar Srivatsa<sup>‡</sup>

Dept of Electrical and Computer Engg, University of Illinois at Urbana-Champaign<sup>†</sup>  
 IBM T.J. Watson Research Center<sup>‡</sup>  
 jizhu1@illinois.edu, msrivats@us.ibm.com

September 22, 2010

## Abstract

Information flow analysis is a powerful technique for reasoning about the sensitive information exposed by a program during its execution. While past work has proposed information theoretic metrics (e.g., Shannon entropy, min-entropy, guessing entropy, etc.) to quantify such information leakage, we argue that some of these measures not only result in counter-intuitive measures of leakage, but also are inherently prone to conflicts when comparing two programs  $P_1$  and  $P_2$  – say Shannon entropy predicts higher leakage for program  $P_1$ , while guessing entropy predicts higher leakage for program  $P_2$ . This paper presents the first attempt towards addressing such conflicts and derives solutions for conflict-free comparison of finite order deterministic programs.

## 1 Introduction

Protecting sensitive and confidential data is becoming more and more important in many fields of human activities, such as electronic commerce, auctions, payments and voting. Information flow analysis is a powerful technique for reasoning about the sensitive information exposed by a program during its execution [1–3]. Existing approaches to information flow analysis can be broadly classified into two: qualitative and quantitative approach. Qualitative information flow analysis, such as taint tracking [4, 5], are coarse-grained – often only distinguishing between *possible* leakage and *no* leakage.

Recently, quantitative information analysis [1, 6–8] techniques have been proposed to alleviate this problem by offering a more fine-grained quantitative assessment of information leakage. Such techniques adopt information theoretic metrics [9, 10] such as mutual information between the secret/sensitive input to a program and its public output to quantify information leakage, as shown in figure 1. In doing so, several entropy measures have been

---

\*A shorter version of this paper is submitted to ICC 2011.

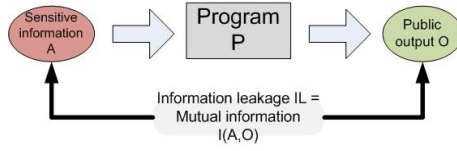


Figure 1: Quantification of Information Leakage in a Program

used to assess mutual information, including, Shannon entropy, Renyi entropy, Guessing entropy (see [6, 7, 11] for more details), and so on. However, in most past work, the choice of such entropy measure has been ad hoc (mostly driven by sample programs) – sometimes leading to counter-intuitive results. Consider the following two programs (by Smith<sup>[7]</sup>), where the secret input  $A$  is uniformly distributed  $8k$ -bit integer with  $k \geq 2$ ,  $\&$  denotes bitwise *and* and  $0^{7k-1}1^{k+1}$  denotes a binary constant.

**PROG P1**

```

if  $A \equiv 0 \pmod{8}$  then
   $O = A$ 
else
   $O = 1$ 
end if

```

**PROG P2**

```

 $O = A \& 0^{7k-1}1^{k+1}$ 

```

Intuitively, one might argue that PROG P1 has much higher information leakage than PROG P2 when  $k$  is large, because it reveals complete information about the secret input with probability  $\frac{1}{8}$ ; on the other hand, when  $k$  is large, PROG P2 reveals roughly  $\frac{1}{8}$  of the number of bits in  $A$ . However, applying Shannon entropy measure and computing the mutual information  $I_1$  between  $A$  and  $O$  yields a counter intuitive result:

$$\begin{aligned}
 P1 : I_1(A, O) &= -\frac{7}{8} \log \frac{7}{8} - \frac{1}{8} \log \frac{1}{2^{8k}} = k + 0.169, \\
 P2 : I_1(A, O) &= -2^{k+1} \cdot \frac{2^{7k-1}}{2^{8k}} \log \frac{2^{7k-1}}{2^{8k}} = k + 1,
 \end{aligned}$$

i.e., leakage by PROG P1 is smaller than leakage by PROG P2, which violates popular consensus in information leakage literature [6, 7]. Indeed, from a security standpoint, PROG P1 leaves  $A$  highly vulnerable to being guessed (e.g., when it is a multiple of 8), while PROG P2 does not (at least for large  $k$ ).

In this paper we argue that past work has failed to address which entropy measure(s) is best suited for quantifying information leakage. Further, this paper shows that some of these entropy based measures (proposed by past work) may be conflicting when they are applied to two programs  $P_1$  and  $P_2$ , i.e., entropy measure  $H$  predicts higher leakage for program  $P_1$ , while entropy measure  $H'$  predicts higher leakage for program  $P_2$ . This paper (to the best of our knowledge) presents the first attempt to analyze different information leakage metrics, show the existence of conflicts in measures proposed by past work and propose a new method for comparing information leakage in finite order deterministic programs.

**Outline.** The paper is structured as follows. In Section 2, we present a program model for finite order deterministic programs. Section 3 shows the existence of conflicts between leakage

measures proposed by past work, followed by our conflict-free leakage metric in Section 4. We analyze a few sample programs using our leakage measure in Section 5 and conclude in Section 6.

## 2 Model Framework

In this section, we present a formal model for a single-input single-output (SISO) deterministic program and Renyi-entropy based definition of information leakage. A SISO deterministic program is modeled as a group of onto mappings:  $O = F_{|\mathcal{A}|}(A), \forall |\mathcal{A}| \in N^+$ , where  $A$  is the high (secret/sensitive) input and  $O$  is the program output, where  $|\mathcal{A}|$  denotes the size of the high input set. In other words, for every  $|\mathcal{A}| \in N^+$ ,  $F_{|\mathcal{A}|}$  is an onto mapping from  $A \in \mathcal{A} = \{0, 1, \dots, |\mathcal{A}| - 1\}$  to  $O \in \mathcal{O}$ . We note that  $|\mathcal{A}|$  acts as a tune able security parameter for the program; assuming  $|\mathcal{O}|$  is fixed, one may be able to increase  $|\mathcal{A}|$  with the goal of improving the security level of the program. More formally, a SISO deterministic program is defined as follows:

**Definition 2.1.** A SISO Deterministic Program is denoted as a 4-tuple  $(\mathbf{q}_{|\mathcal{A}|}, |\mathcal{A}|, F_{|\mathcal{A}|}, \mathbf{p}_{|\mathcal{A}|})$ , where  $\forall |\mathcal{A}| \in N^+$ ,  $A$  is a random variable in  $\mathcal{A} = \{0, 1, \dots, |\mathcal{A}| - 1\}$  with distribution vector  $\mathbf{q}_{|\mathcal{A}|}$ ,  $O = F_{|\mathcal{A}|}(A)$  is an onto mapping from  $\mathcal{A}$  to  $\mathcal{O}$ , and  $\mathbf{p}_{|\mathcal{A}|}$  denotes the distribution vector of output  $O$  under mapping  $F_{|\mathcal{A}|}(\cdot)$ .

A SISO deterministic program  $(\mathbf{q}_{|\mathcal{A}|}, |\mathcal{A}|, F_{|\mathcal{A}|}, \mathbf{p}_{|\mathcal{A}|})$  is said to be a Finite Order SISO Deterministic Program (FOP) if and only if

$$\sup_{|\mathcal{A}| \in N^+} \|\mathbf{p}_{|\mathcal{A}|}\|_0 < \infty$$

It is called an Infinite Order SISO Deterministic Program (IOP) if and only if

$$\sup_{|\mathcal{A}| \in N^+} \|\mathbf{p}_{|\mathcal{A}|}\|_0 = \infty$$

where  $\|\mathbf{p}_{|\mathcal{A}|}\|_0$  is the zero norm of  $\mathbf{p}_{|\mathcal{A}|}$ .

Unless explicitly specified, in the following portions of this paper, we assume that the secret input  $A$  has an uniform prior distribution in  $\mathcal{A}$  for any  $|\mathcal{A}|$ .

A key difference between FOPs and IOPs is that the entropy of output  $O$  is bounded for FOPs, and so is information leakage. Assuming that  $|\mathcal{O}|$  is fixed (independent of  $|\mathcal{A}|$ ), intuitively the security level of a real FOP will be non-decreasing in  $|\mathcal{A}|$ . In the following portions of this paper we focus on information leakage metrics for FOPs.

Having formalized the program model, we define leakage using Renyi entropy [12], which covers most of the entropy metrics adopted by past work on information flow analysis [6, 7, 11, 13], such as Shannon entropy, min-entropy, vulnerability one-guess entropy (proposed by Hamadou et. al, [6]), etc. Renyi entropy is defined as follows: For a random variable  $X$  with distribution  $\mathbf{p} = (p_0, p_1, \dots, p_n)$ , its Renyi entropy is defined as:

$$H_\alpha(X) = \frac{1}{1 - \alpha} \log \sum_{i=0}^n p_i^\alpha$$

where  $\alpha$  is a parameter. In this paper we also apply  $H_\alpha(\mathbf{p})$  to denote  $H_\alpha(X)$ . When  $\alpha = 1$ , Renyi entropy becomes Shannon entropy; when  $\alpha \rightarrow \infty$ ,  $H_\infty(X) = -\log \sup_i p_i$  is the min-entropy; when  $\alpha = 0$ ,  $H_0(X)$  denotes the vulnerability one-guess entropy.

According to general consensus in information flow analysis literature, information leakage ( $IL$ ) of a program  $C = (\mathbf{q}_{|\mathcal{A}|}, |\mathcal{A}|, F_{|\mathcal{A}|}, \mathbf{p}_{|\mathcal{A}|})$  (at a given  $|\mathcal{A}|$ ) under  $\alpha$ -Renyi entropy metric is defined as the mutual information  $I_\alpha$  between  $O$  and  $A$ :

$$IL_\alpha(C, |\mathcal{A}|) = I_\alpha(O, A) = H_\alpha(O) - H_\alpha(O|A) = H_\alpha(O)$$

where  $IL_\alpha(C, |\mathcal{A}|)$  denotes a class of information leakage metrics (for different values of  $\alpha$ ) of program  $C$ . Note that since the program is deterministic  $H_\alpha(O|A) = 0, \forall \alpha$ .

It is worth noting that the mutual information  $I_\alpha(O, A)$  may also be defined as  $I_\alpha(O, A) = H_\alpha(A) - H_\alpha(A|O)$ , which differs from  $H_\alpha(O) - H_\alpha(O|A)$  when  $\alpha \neq 1$ . This alternative definition is not considered here because when  $A$  is uniformly distributed,  $I_\alpha(O, A) = H_\alpha(A) - H_\alpha(A|O)$  reduces to be Shannon mutual information for all  $\alpha$ , as shown below:

$$\begin{aligned} H_\alpha(A) - H_\alpha(A|O) &= -\log |\mathcal{A}| - \sum_{o \in \mathcal{O}} P(O = o) H_\alpha(A|O = o) \\ &= -\log |\mathcal{A}| + \sum_{o \in \mathcal{O}} P(O = o) \log |\{a : F_{|\mathcal{A}|}(a) = o\}| \\ &= -\log |\mathcal{A}| + \sum_{o \in \mathcal{O}} P(O = o) \log (|\mathcal{A}| P(O = o)) = H_1(O) \end{aligned}$$

In the next section, we show that this definition of information leakage results in conflicts when comparing two programs. In the subsequent sections we develop solutions for conflict-free comparison of two programs.

### 3 Conflicts in Information Leakage metrics

In this section we show several examples of conflicts while comparing two program's information leakage. Recall PROG P1 and PROG P2 from Section 1. Consider the Renyi mutual information of these two PROGs when  $\alpha = 0, 1, \infty$ .

$$\begin{cases} IL_0(P1, 2^{8k}) = 8k - 3, & IL_0(P2, 2^{8k}) = k + 1 \\ IL_1(P1, 2^{8k}) = k + 0.169, & IL_1(P2, 2^{8k}) = k + 1 \\ IL_\infty(P1, 2^{8k}) = 0.134, & IL_\infty(P2, 2^{8k}) = k + 1 \end{cases}$$

Note that only the comparing  $IL_0(P1, 2^{8k})$  and  $IL_0(P2, 2^{8k})$  agrees with our intuition that P1 leaks much more information than P2; however, comparing  $IL_1(P1, 2^{8k})$  and  $IL_1(P2, 2^{8k})$  shows that P1 leaks about the same amount of information as P2; comparing  $IL_\infty(P1, 2^{8k})$  and  $IL_\infty(P2, 2^{8k})$  shows that P2 leaks much more information than P1. We see that the leakage measures for different  $\alpha$  values conflict with each other, and some of them are even counter-intuitive.

Smith [7] and Hamadou et. al. [6] argue that  $IL_0$  is more important than  $IL_1$  in information flow analysis, because in the above example,  $IL_0$  coincides with the intuition but  $IL_1$

does not. However, it is not difficult to come up with other examples where  $IL_1$  coincides with the intuition but  $IL_0$  does not. Consider the following two programs, where the high input  $A$  is an uniformly distributed  $k$ -bit integer with  $k \geq 2$  and  $L$  is a parameter in  $\mathcal{A}$ .

**PROG P3** Password Checker

```

if  $A = L$  then
   $O = 1$ 
else
   $O = 0$ 
end if

```

**PROG P4** Binary Search

```

if  $A \geq L$  then
   $O = 1$ 
else
   $O = 0$ 
end if

```

Consider  $L = |\mathcal{A}|/2$ . The intuition is that PROG P4 leaks much more information than PROG P3, because when  $k$  is large, the probability of  $A = L$  becomes so low that PROG P3 leaks almost no information. But PROG P4 always leaks 1 bit of information, irrespective of  $|\mathcal{A}|$ . Now, consider the Renyi mutual information when  $\alpha = 0, 1, \infty$ :

$$\begin{cases} IL_0(P3, 2^k) = 1, IL_0(P4, 2^k) = 1 \\ IL_1(P3, 2^k) = H_1(\frac{|\mathcal{A}|-1}{|\mathcal{A}|}, \frac{1}{|\mathcal{A}|}), IL_1(P4, 2^k) = 1 \\ IL_\infty(P3, 2^k) = -\log(1 - \frac{1}{|\mathcal{A}|}), IL_\infty(P4, 2^k) = 1 \end{cases}$$

We see that the comparing result when  $\alpha = 0$  fails to coincide with the intuition, while the comparing results when  $\alpha = 1$  or  $\infty$  match the intuition. The conflict between information leakage metrics for different values of  $\alpha$  appears again.

The following lemma indicates that the conflict between different metrics is very common.

**Lemma 3.1.**  $\forall \alpha \geq 0, \beta \geq 0, \alpha \neq \beta$ , there exists two SISO deterministic programs  $C_1 = (\mathbf{q}_{|\mathcal{A}|}, |\mathcal{A}|, F_{|\mathcal{A}|}, \mathbf{p}_{|\mathcal{A}|})$  and  $C_2 = (\mathbf{q}'_{|\mathcal{A}|}, |\mathcal{A}|, F'_{|\mathcal{A}|}, \mathbf{p}'_{|\mathcal{A}|})$  with  $\mathbf{q}_{|\mathcal{A}|}$  and  $\mathbf{q}'_{|\mathcal{A}|}$  both being uniform distributions in  $\mathcal{A}$ , such that  $\exists D \in R^+$ , if  $|\mathcal{A}| > D$ ,

$$IL_\alpha(C_1, |\mathcal{A}|) > IL_\alpha(C_2, |\mathcal{A}|) \tag{1}$$

$$IL_\beta(C_1, |\mathcal{A}|) < IL_\beta(C_2, |\mathcal{A}|) \tag{2}$$

*Proof.* The key idea to construct the programs stem from the following property of Renyi entropy  $H_\alpha(\mathbf{p})$ :  $H_\alpha(\mathbf{p})$  is a monotone decreasing function of  $\alpha$  for any specific  $\mathbf{p}$ . Moreover, if  $\mathbf{p}$  is uniform,  $H_\alpha(\mathbf{p})$  is a constant (independent of  $\alpha$ ); if  $\mathbf{p}$  contains a peak probability and a large number of small probabilities,  $H_\alpha(\mathbf{p})$  will decreasing quickly as  $\alpha$  increases (see [12] for details).

First, let us suppose  $1 < \alpha < \beta \leq \infty$ . Pick values  $p_0 \in (0, 1), n \in N^+$  such that

$$\frac{1}{2^{1-1/\beta}} < p_0 < \frac{1}{2^{1-1/\alpha}} \quad (3)$$

$$\log \left[ p_0^\beta + \frac{(1-p_0)^\beta}{n^{\beta-1}} \right] > 1 - \beta \quad (4)$$

$$\log \left[ p_0^\alpha + \frac{(1-p_0)^\alpha}{n^{\alpha-1}} \right] < 1 - \alpha \quad (5)$$

We note that one can first pick  $p_0$  satisfying (3); then, to satisfy (4) and (5) one simply needs to choose a sufficiently large value for  $n$ .

Specify the mapping function  $F_{|\mathcal{A}|}$  for  $C_1$  so that the distribution of  $O$  is  $\mathbf{p}_{|\mathcal{A}|} = (p_0, p_1, \dots, p_n)$  with  $p_0$  chosen as described above and  $p_1 = \dots = p_n = \frac{1-p_0}{n}$  for any  $|\mathcal{A}| > n+1$ , and specify the mapping function  $F'_{|\mathcal{A}|}$  for  $C_2$  so that the distribution of  $O'$  is  $\mathbf{p}'_{|\mathcal{A}|} = (1/2, 1/2)$  for any  $|\mathcal{A}|$ . Then, for any  $|\mathcal{A}| > n+1$ ,

$$IL_\alpha(C_1, |\mathcal{A}|) = H_\alpha(\mathbf{p}_{|\mathcal{A}|}) = \frac{1}{1-\alpha} \log \left[ p_0^\alpha + \frac{(1-p_0)^\alpha}{n^{\alpha-1}} \right] > 1 = H_\alpha(\mathbf{p}'_{|\mathcal{A}|}) = IL_\alpha(C_2, |\mathcal{A}|),$$

$$IL_\beta(C_1, |\mathcal{A}|) = H_\beta(\mathbf{p}_{|\mathcal{A}|}) = \frac{1}{1-\beta} \log \left[ p_0^\beta + \frac{(1-p_0)^\beta}{n^{\beta-1}} \right] < 1 = H_\beta(\mathbf{p}'_{|\mathcal{A}|}) = IL_\beta(C_2, |\mathcal{A}|),$$

equations (1) and (2) are satisfied.

In the case that  $1 < \beta < \alpha \leq \infty$ , switch the mapping function of  $F_{|\mathcal{A}|}$  and  $F'_{|\mathcal{A}|}$  above, so that the distribution of  $O_{|\mathcal{A}|}$  is  $\mathbf{q}$  and the distribution of  $O'_{|\mathcal{A}|}$  is  $\mathbf{p}$ , then (1) and (2) are still valid.

Second, suppose  $0 \leq \alpha < \beta < 1$ , pick  $2 \leq m, n \in N^+$  so that

$$\frac{1}{1-\alpha} \log \left[ \left(\frac{1}{2}\right)^\alpha + \left(\frac{1}{2}\right)^\alpha n^{1-\alpha} \right] > \log m > \frac{1}{1-\beta} \log \left[ \left(\frac{1}{2}\right)^\beta + \left(\frac{1}{2}\right)^\beta n^{1-\beta} \right] \quad (6)$$

A sufficiently large  $n$  for (6) can make it possible to choose a valid  $m$ .

Specify the mapping function  $F_{|\mathcal{A}|}$  for  $C_1$  so that the distribution of  $O$  is  $\mathbf{p}_{|\mathcal{A}|} = (p_0, p_1, \dots, p_n)$  with  $p_0 = \frac{1}{2}$  and  $p_1 = \dots = p_n = \frac{1}{2n}$  for any  $|\mathcal{A}| > n+1$ , and specify the mapping function  $F'_{|\mathcal{A}|}$  for  $C_2$  so that the distribution of  $O'$  is  $\mathbf{p}'_{|\mathcal{A}|} = (1/m, 1/m, \dots, 1/m)$  for any  $|\mathcal{A}|$ . Then, for any  $|\mathcal{A}| > n+1$ , equations (1) and (2) are satisfied. The case of  $0 \leq \beta < \alpha < 1$  can be proved by switching  $F_{|\mathcal{A}|}$  and  $F'_{|\mathcal{A}|}$  as done before.

Third, suppose  $\alpha$  and  $\beta$  belong to  $[0, 1]$  and  $[1, \infty]$  separately. For example, if  $\alpha < 1 \leq \beta$ , pick a value  $\beta'$  such that  $\alpha < \beta' < 1$ , and construct  $C_1$  and  $C_2$  by the same method above, with  $\beta'$  in place of  $\beta$ . Then (1) and (2) can be satisfied because  $IL_{\beta'}(C_2, |\mathcal{A}|) = IL_\beta(C_2, |\mathcal{A}|) = IL_\alpha(C_2, |\mathcal{A}|)$  as  $\mathbf{p}'_{|\mathcal{A}|}$  is uniform. Equations (1) and (2) in other case of  $\alpha$  and  $\beta$  can be justified in the same way. □

## 4 Quantifying Information Leakage in FOPs

So far we have shown that some measures of information leakage are not only counter-intuitive, but also introduce conflicts when comparing two programs. In this section we develop a new

approach to quantify and compare information leakage in programs. We first sketch the key idea behind our approach. Recall that in FOPs,  $|\mathcal{A}|$  acts as a security parameter for the program – intuitively, increasing  $|\mathcal{A}|$  increases the security level of the program (since,  $|\mathcal{O}|$  is finite and constant – independent of  $|\mathcal{A}|$ ). Recall the password checker PROG P3 – observe that increasing the length of the password ( $A$ ) by one bit doubles the security level of the program.

In this paper we propose that two FOPs  $C_1$  and  $C_2$  should be compared by examining  $\lim_{|\mathcal{A}| \rightarrow \infty} IL_\alpha(C_1, |\mathcal{A}|) \lim_{|\mathcal{A}| \rightarrow \infty} IL_\alpha(C_2, |\mathcal{A}|)$ . In particular, we show that one can obtain conflict free comparison of programs using a relative leakage metric defined by the ratio  $\lim_{|\mathcal{A}| \rightarrow \infty} \frac{IL_\alpha(C_1, |\mathcal{A}|)}{IL_\alpha(C_2, |\mathcal{A}|)}$ . Evidently, if the relative leakage metric is 0, then program  $C_2$  leaks more information than program  $C_1$ ; if the relative leakage metric is  $\infty$ , then program  $C_1$  leaks more information than program  $C_2$ . Now, if the relative leakage metric of programs  $C_1$  and  $C_2$  is a constant  $c$  ( $c \neq 0, \infty$ ), one may increase the size of the secret input (namely,  $\log |\mathcal{A}|$ ) for program  $C_1$  by a constant factor relative to the size of the secret input for program  $C_2$  to ensure that the programs  $C_1$  and  $C_2$  have equal security level; hence, in this case we conclude that the programs  $C_1$  and  $C_2$  are equal with respect to information leakage. In this section, we formalize this intuition and present a conflict-free approach to comparing information leakage in FOPs.

We first show that for any  $C = (\mathbf{q}_{|\mathcal{A}|}, |\mathcal{A}|, F_{|\mathcal{A}|}, \mathbf{p}_{|\mathcal{A}|})$ ,  $IL_\alpha(C, |\mathcal{A}|)$  is closely related to  $\|\mathbf{p}_{|\mathcal{A}|}\|_\infty$ .

**Lemma 4.1.**  $\forall 2 \leq n \in N$ , for any probability distribution vector  $\mathbf{p} = (p_1, p_2, \dots, p_n)$  with ordered sequence  $\|\mathbf{p}\|_\infty = p_1 \geq p_2 \geq \dots \geq p_n$ , then,

$$\forall 1 < \alpha \leq \infty, \lim_{p_1 \rightarrow 1} \frac{H_\alpha(\mathbf{p})}{1 - p_1} = \frac{\alpha}{\alpha - 1} \quad (7)$$

$$\lim_{p_1 \rightarrow 1} \frac{H_1(\mathbf{p})}{-(1 - p_1) \log(1 - p_1)} = 1 \quad (8)$$

$$\forall \alpha \in (0, 1), \begin{cases} \liminf_{p_1 \rightarrow 1} \frac{H_\alpha(\mathbf{p})}{(1 - p_1)^\alpha} > 0 \\ \limsup_{p_1 \rightarrow 1} \frac{H_\alpha(\mathbf{p})}{(1 - p_1)^\alpha} < \infty \end{cases} \quad (9)$$

*Proof.* Consider (7), use substitution  $t = 1 - p_1$ . When  $\alpha = \infty$ ,

$$\lim_{p_1 \rightarrow 1} \frac{H_\infty(\mathbf{p})}{1 - p_1} = \lim_{t \rightarrow 0} \frac{-\log(1 - t)}{t} = 1$$

When  $1 < \alpha < \infty$ , note that  $\forall \mathbf{p}$  with  $n \geq 2$ ,

$$\frac{1}{(n - 1)^{\alpha - 1}} \leq \sum_{i=2}^n (p_i/t)^\alpha \leq 1,$$

so we have

$$\begin{aligned}
\lim_{p_1 \rightarrow 1} \frac{H_\alpha(\mathbf{p})}{1 - p_1} &= \frac{1}{1 - \alpha} \lim_{t \rightarrow 0} \frac{(1 - t)^\alpha - 1 + \sum_{i=2}^n p_i^\alpha}{t} \\
&= \frac{1}{1 - \alpha} \left( -\alpha + \lim_{t \rightarrow 0} t^{\alpha-1} \sum_{i=2}^n (p_i/t)^\alpha \right) \\
&= \frac{\alpha}{\alpha - 1}
\end{aligned}$$

Thus, equation (7) holds. Next consider (8).

When  $\alpha = 1$ , note that  $\forall \mathbf{p}$  with  $n \geq 2$ ,

$$0 \leq \left| \sum_{i=2}^n \frac{p_i}{t} \log \frac{p_i}{t} \right| \leq \log(n - 1),$$

and we have

$$\begin{aligned}
&\lim_{p_1 \rightarrow 1} \frac{H_1(\mathbf{p})}{(1 - p_1) \log(1 - p_1)} \\
&= \lim_{t \rightarrow 0} \frac{(1 - t) \log(1 - t) + \sum_{i=2}^n p_i \log p_i}{-t \log t} \\
&= 1 - \lim_{t \rightarrow 0} \frac{1}{\log t} \sum_{i=2}^n \left( \frac{p_i}{t} \log \frac{p_i}{t} \right) = 1
\end{aligned}$$

Thus, equation (8) holds. Next consider (9).

When  $0 < \alpha < 1$ , note that  $\forall \mathbf{p}$  with  $n \geq 2$ ,

$$1 \leq \sum_{i=2}^n (p_i/t)^\alpha \leq (n - 1)^{1-\alpha},$$

so we have

$$\begin{aligned}
&\limsup_{p_1 \rightarrow 1} \frac{H_\alpha(\mathbf{p})}{(1 - p_1)^\alpha} \\
&\leq \frac{1}{1 - \alpha} \left( \lim_{t \rightarrow 0} -\alpha t^{1-\alpha} + \limsup_{t \rightarrow 0} \sum_{i=2}^n (p_i/t)^\alpha \right) \\
&= \frac{1}{1 - \alpha} \limsup_{t \rightarrow 0} \sum_{i=2}^n (p_i/t)^\alpha \leq \frac{(n - 1)^{1-\alpha}}{1 - \alpha} < \infty.
\end{aligned}$$

The other part of (9) can be proved in the same way.  $\square$

Define a function  $T_\alpha(\cdot)$  for random distributions  $\mathbf{p} = (p_1, p_2, \dots, p_n)$  with  $p_1 \geq p_2 \geq \dots \geq p_n$ :

$$T_\alpha(\mathbf{p}) = \begin{cases} 1 - p_1, & \text{if } \alpha > 1 \\ -[1 - p_1] \log[1 - p_1], & \text{if } \alpha = 1 \\ [1 - p_1]^\alpha, & \text{if } 0 < \alpha < 1 \end{cases} \quad (10)$$



Lemma 4.1 shows that  $\forall \alpha > 0, \lim_{\|\mathbf{p}\|_\infty \rightarrow 1} \frac{H_\alpha(\mathbf{p})}{T_\alpha(\mathbf{p})}$  is finite. Further, if  $2 \leq n$  is finite and  $\frac{1}{n} \leq \|\mathbf{p}\|_\infty < 1 - \epsilon$  for some  $\epsilon > 0$ , both  $H_\alpha(\mathbf{p})$  and  $T_\alpha(\mathbf{p})$  will be upper bounded by  $\log n < \infty$  and will both be strictly larger than zero. This leads us to Proposition 4.2.

**Proposition 4.2.** *For any FOP  $C = \{\mathbf{q}_{|\mathcal{A}|}, |\mathcal{A}|, F_{|\mathcal{A}|}, \mathbf{p}_{|\mathcal{A}|}\}$  with  $\mathbf{q}_{|\mathcal{A}|}$  being uniform in  $\mathcal{A}$ , we have*

$$\forall \alpha > 0, 0 < \inf_{|\mathcal{A}|} \frac{IL_\alpha(C, |\mathcal{A}|)}{T_\alpha(\mathbf{p}_{|\mathcal{A}|})} \leq \sup_{|\mathcal{A}|} \frac{IL_\alpha(C, |\mathcal{A}|)}{T_\alpha(\mathbf{p}_{|\mathcal{A}|})} < \infty$$

Proposition 4.2 states that as  $\alpha$  is varied, the values of  $IL_\alpha$  differ among the levels:  $1 - \|\mathbf{p}_{|\mathcal{A}|}\|_\infty$ ,  $(1 - \|\mathbf{p}_{|\mathcal{A}|}\|_\infty) \log(1 - \|\mathbf{p}_{|\mathcal{A}|}\|_\infty)$ ,  $(1 - \|\mathbf{p}_{|\mathcal{A}|}\|_\infty)^\alpha$ . Note that these levels are all related to  $1 - \|\mathbf{p}_{|\mathcal{A}|}\|_\infty$ . Intuitively, the rate of convergence of  $1 - \|\mathbf{p}_{|\mathcal{A}|}\|_\infty$  to 0 determines the security level of a program. We formalize this notion in the following proposition:

**Proposition 4.3.** *For any FOPs  $C_1 = (\mathbf{q}_{|\mathcal{A}|}, |\mathcal{A}|, F_{|\mathcal{A}|}, \mathbf{p}_{|\mathcal{A}|})$  and  $C_2 = (\mathbf{q}'_{|\mathcal{A}|}, |\mathcal{A}|, F'_{|\mathcal{A}|}, \mathbf{p}'_{|\mathcal{A}|})$ , with  $\mathbf{q}_{|\mathcal{A}|}$  and  $\mathbf{q}'_{|\mathcal{A}|}$  both being uniform in  $\mathcal{A}$ . Applying notation*

$$f_\alpha = \limsup_{|\mathcal{A}| \rightarrow \infty} \frac{IL_\alpha(C_1, |\mathcal{A}|)}{IL_\alpha(C_2, |\mathcal{A}|)}, \quad g_\alpha = \liminf_{|\mathcal{A}| \rightarrow \infty} \frac{IL_\alpha(C_1, |\mathcal{A}|)}{IL_\alpha(C_2, |\mathcal{A}|)},$$

we have:

1.

$$\begin{cases} \exists \alpha > 0, f_\alpha = 0 \Leftrightarrow \forall \beta > 0, f_\beta = 0 \\ \exists \alpha > 0, f_\alpha = \infty \Leftrightarrow \forall \beta > 0, f_\beta = \infty \\ \exists \alpha > 0, 0 < f_\alpha < \infty \Leftrightarrow \forall \beta > 0, 0 < f_\beta < \infty \end{cases} \quad (11)$$

2.

$$\begin{cases} \exists \alpha > 0, g_\alpha = 0 \Leftrightarrow \forall \beta > 0, g_\beta = 0 \\ \exists \alpha > 0, g_\alpha = \infty \Leftrightarrow \forall \beta > 0, g_\beta = \infty \\ \exists \alpha > 0, 0 < g_\alpha < \infty \Leftrightarrow \forall \beta > 0, 0 < g_\beta < \infty \end{cases} \quad (12)$$

*Proof.* It follows directly from Proposition 4.2 that for any  $\alpha > 0$ ,

$$\begin{aligned} f_\alpha = 0 &\Leftrightarrow \limsup_{|\mathcal{A}| \rightarrow \infty} \frac{T_\alpha(\mathbf{p}_{|\mathcal{A}|})}{T_\alpha(\mathbf{p}'_{|\mathcal{A}|})} = 0, \quad f_\alpha = \infty \Leftrightarrow \limsup_{|\mathcal{A}| \rightarrow \infty} \frac{T_\alpha(\mathbf{p}_{|\mathcal{A}|})}{T_\alpha(\mathbf{p}'_{|\mathcal{A}|})} = \infty \\ 0 < f_\alpha < \infty &\Leftrightarrow 0 < \limsup_{|\mathcal{A}| \rightarrow \infty} \frac{T_\alpha(\mathbf{p}_{|\mathcal{A}|})}{T_\alpha(\mathbf{p}'_{|\mathcal{A}|})} < \infty \\ g_\alpha = 0 &\Leftrightarrow \liminf_{|\mathcal{A}| \rightarrow \infty} \frac{T_\alpha(\mathbf{p}_{|\mathcal{A}|})}{T_\alpha(\mathbf{p}'_{|\mathcal{A}|})} = 0, \quad g_\alpha = \infty \Leftrightarrow \liminf_{|\mathcal{A}| \rightarrow \infty} \frac{T_\alpha(\mathbf{p}_{|\mathcal{A}|})}{T_\alpha(\mathbf{p}'_{|\mathcal{A}|})} = \infty \\ 0 < g_\alpha < \infty &\Leftrightarrow 0 < \liminf_{|\mathcal{A}| \rightarrow \infty} \frac{T_\alpha(\mathbf{p}_{|\mathcal{A}|})}{T_\alpha(\mathbf{p}'_{|\mathcal{A}|})} < \infty \end{aligned}$$

Note that for any intervals  $T, S \subset (0, 1)$  and any variables  $t \in T, s \in S, \forall v \in \{0, \infty\}$ ,

$$\begin{aligned} \limsup_{t \in T, s \in S} \frac{1-t}{1-s} = v &\Leftrightarrow \limsup_{t \in T, s \in S} \frac{(1-t) \log(1-t)}{(1-s) \log(1-s)} = v \Leftrightarrow \limsup_{t \in T, s \in S} \frac{(1-t)^\beta}{(1-s)^\beta} = v, \forall \beta \in (0, 1), \\ \liminf_{t \in T, s \in S} \frac{1-t}{1-s} = v &\Leftrightarrow \liminf_{t \in T, s \in S} \frac{(1-t) \log(1-t)}{(1-s) \log(1-s)} = v \Leftrightarrow \liminf_{t \in T, s \in S} \frac{(1-t)^\beta}{(1-s)^\beta} = v, \forall \beta \in (0, 1), \end{aligned}$$

which indicates that,

$$\begin{aligned} \exists \alpha > 0, \limsup_{|\mathcal{A}| \rightarrow \infty} \frac{T_\alpha(\mathbf{p}_{|\mathcal{A}|})}{T_\alpha(\mathbf{p}'_{|\mathcal{A}|})} = v &\Leftrightarrow \forall \beta > 0, \limsup_{|\mathcal{A}| \rightarrow \infty} \frac{T_\alpha(\mathbf{p}_{|\mathcal{A}|})}{T_\alpha(\mathbf{p}'_{|\mathcal{A}|})} = v \\ \exists \alpha > 0, 0 < \limsup_{|\mathcal{A}| \rightarrow \infty} \frac{T_\alpha(\mathbf{p}_{|\mathcal{A}|})}{T_\alpha(\mathbf{p}'_{|\mathcal{A}|})} < \infty &\Leftrightarrow \forall \beta > 0, 0 < \limsup_{|\mathcal{A}| \rightarrow \infty} \frac{T_\alpha(\mathbf{p}_{|\mathcal{A}|})}{T_\alpha(\mathbf{p}'_{|\mathcal{A}|})} < \infty \\ \exists \alpha > 0, \liminf_{|\mathcal{A}| \rightarrow \infty} \frac{T_\alpha(\mathbf{p}_{|\mathcal{A}|})}{T_\alpha(\mathbf{p}'_{|\mathcal{A}|})} = v &\Leftrightarrow \forall \beta > 0, \liminf_{|\mathcal{A}| \rightarrow \infty} \frac{T_\alpha(\mathbf{p}_{|\mathcal{A}|})}{T_\alpha(\mathbf{p}'_{|\mathcal{A}|})} = v \\ \exists \alpha > 0, 0 < \liminf_{|\mathcal{A}| \rightarrow \infty} \frac{T_\alpha(\mathbf{p}_{|\mathcal{A}|})}{T_\alpha(\mathbf{p}'_{|\mathcal{A}|})} < \infty &\Leftrightarrow \forall \beta > 0, 0 < \liminf_{|\mathcal{A}| \rightarrow \infty} \frac{T_\alpha(\mathbf{p}_{|\mathcal{A}|})}{T_\alpha(\mathbf{p}'_{|\mathcal{A}|})} < \infty \end{aligned}$$

So we conclude that (11) and (12) are valid.  $\square$

Now, we are ready to present our solution to compare information leakage of two programs:

**Algorithm 4.4.** For any FOPs  $C_1 = (\mathbf{q}_{|\mathcal{A}|}, |\mathcal{A}|, F_{|\mathcal{A}|}, \mathbf{p}_{|\mathcal{A}|})$  and  $C_2 = (\mathbf{q}'_{|\mathcal{A}|}, |\mathcal{A}|, F'_{|\mathcal{A}|}, \mathbf{p}'_{|\mathcal{A}|})$ , with  $\mathbf{q}_{|\mathcal{A}|}$  and  $\mathbf{q}'_{|\mathcal{A}|}$  both being uniform in  $\mathcal{A}$ ,

*BEGIN PROGRAM*

**if**  $f_\infty = \infty$  and  $g_\infty > 0$  **then**

$C_1$  has a higher leakage than  $C_2$ .

**else if**  $f_\infty < \infty$  and  $g_\infty = 0$  **then**

$C_2$  has a higher leakage than  $C_1$

**else if**  $0 < g_\infty \leq f_\infty < \infty$  **then**

$C_1$  and  $C_2$  are on the same leakage level.

**else**

$C_1$  and  $C_2$  are not comparable.

**end if**

*END PROGRAM.*

If  $\frac{IL_\infty(C_1, |\mathcal{A}|)}{IL_\infty(C_2, |\mathcal{A}|)}$  converges as  $|\mathcal{A}| \rightarrow \infty$ , Algorithm 4.4 can be rewritten as:

**Algorithm 4.5.**

*BEGIN PROGRAM*

**if**  $\lim_{|\mathcal{A}| \rightarrow \infty} \frac{IL_\infty(C_1, |\mathcal{A}|)}{IL_\infty(C_2, |\mathcal{A}|)} = \infty$  **then**

$C_1$  has a higher leakage than  $C_2$

**else if**  $\lim_{|\mathcal{A}| \rightarrow \infty} \frac{IL_\infty(C_1, |\mathcal{A}|)}{IL_\infty(C_2, |\mathcal{A}|)} = 0$  **then**

$C_2$  has a higher leakage than  $C_1$

**else**

$C_1$  and  $C_2$  are on the same leakage level.  
**end if**  
**END PROGRAM.**

Due to Algorithm 4.5, it is natural to define the leakage level of a FOP  $C$  as the rate of convergence of  $IL_\infty(C, |\mathcal{A}|)$  as  $|\mathcal{A}| \rightarrow \infty$ :

**Definition 4.6. Leakage Level** For any FOPs  $C = (\mathbf{q}_{|\mathcal{A}|}, |\mathcal{A}|, F_{|\mathcal{A}|}, \mathbf{p}_{|\mathcal{A}|})$  with  $\mathbf{q}_{|\mathcal{A}|}$  being uniform in  $|\mathcal{A}|$ , if  $IL_\infty(C, |\mathcal{A}|)$  converges as  $|\mathcal{A}| \rightarrow \infty$ , then the leakage level of  $C$  is defined to be  $\Theta^{[14]}(IL_\infty(C, |\mathcal{A}|)) = \Theta(1 - \|\mathbf{p}_{|\mathcal{A}|}\|_\infty)$ .

We claim that algorithm 4.4 (and thus algorithm 4.5) offers a conflict-free solution to comparing information leakage of two programs. The proof follows directly from Proposition 4.3. We note that in algorithm 4.4 that there may be cases wherein two programs are incomparable. However, we claim that it may be impossible to offer a more fine-grained comparison of two programs using Renyi-entropy measure as follows. First, we observe that in Algorithm 4.4, information leakage measures for two are distinguishable if and only if the ratio of their min-entropy leakage metric is either  $0 = 1/\infty$  or  $\infty$ . The following lemma shows that it is impossible to reduce this ratio to some finite  $D < \infty$ :

**Lemma 4.7.**  $\forall D > 1, \exists \alpha, \beta \in (0, \infty], \alpha \neq \beta, \exists$  FOPs  $C_1 = (\mathbf{q}_{|\mathcal{A}|}, |\mathcal{A}|, F_{|\mathcal{A}|}, \mathbf{p}_{|\mathcal{A}|})$  and  $C_2 = (\mathbf{q}'_{|\mathcal{A}|}, |\mathcal{A}|, F'_{|\mathcal{A}|}, \mathbf{p}'_{|\mathcal{A}|})$ , with  $\mathbf{q}_{|\mathcal{A}|}$  and  $\mathbf{q}'_{|\mathcal{A}|}$  both being uniform in  $\mathcal{A}$ , such that,

$$\begin{aligned} \lim_{|\mathcal{A}| \rightarrow \infty} \frac{IL_\alpha(C_1, |\mathcal{A}|)}{IL_\alpha(C_2, |\mathcal{A}|)} &> D \text{ but} \\ \lim_{|\mathcal{A}| \rightarrow \infty} \frac{IL_\beta(C_1, |\mathcal{A}|)}{IL_\beta(C_2, |\mathcal{A}|)} &< \frac{1}{D} \end{aligned}$$

*Proof.* We first give an intuitive explanation of the proof of Lemma 4.7 here. Recall from Lemma 4.1 that it is feasible to make  $\frac{H_\alpha(\mathbf{p})}{H_\beta(\mathbf{p})}$  as large as possible for distribution  $\mathbf{p}$  with  $\|\mathbf{p}\|_\infty$  close enough to 1. This allows us to construct a program  $C_1$  with  $\|\mathbf{p}_{|\mathcal{A}|}\|_\infty$  close to 1, so that we have  $IL_\alpha(C_1, |\mathcal{A}|)/IL_\beta(C_1, |\mathcal{A}|) > D^2$  (when  $|\mathcal{A}|$  is large) and a program  $C_2$  with  $IL_\alpha(C_2, |\mathcal{A}|) = IL_\beta(C_2, |\mathcal{A}|) = \sqrt{IL_\alpha(C_1, |\mathcal{A}|)IL_\beta(C_1, |\mathcal{A}|)}$  (when  $|\mathcal{A}|$  is large). Clearly, the constructed programs  $C_1$  and  $C_2$  satisfies Lemma 4.7.

Here we offer a simple example of  $C_1$  and  $C_2$ . Choose  $p_0 \in (0, 1), 2 \leq n \in \mathbb{N}$  such that,

$$\begin{aligned} 2^{-1/D} &< p_0 < 1 \\ \log n &> D - \frac{\alpha}{1-\alpha} \log(1-p_0) \end{aligned}$$

Specify  $C_1$  so that  $\mathbf{p}_{|\mathcal{A}|} = (p_0, \frac{1-p_0}{n}, \frac{1-p_0}{n}, \dots, \frac{1-p_0}{n})$  for any  $|\mathcal{A}| > n+1$ , and specify  $C_2$  so that  $\mathbf{p}'_{|\mathcal{A}|} = (1/2, 1/2)$  for any  $|\mathcal{A}|$ . And consider  $0 < \alpha < 1, \beta = \infty$ , then

$$\begin{aligned} \lim_{|\mathcal{A}| \rightarrow \infty} \frac{IL_\alpha(C_1, |\mathcal{A}|)}{IL_\alpha(C_2, |\mathcal{A}|)} &= \frac{1}{1-\alpha} \log [p_0^\alpha + (1-p_0)^\alpha n^{1-\alpha}] \geq \frac{1}{1-\alpha} \log [(1-p_0)^\alpha n^{1-\alpha}] > D \\ \lim_{|\mathcal{A}| \rightarrow \infty} \frac{IL_\beta(C_1, |\mathcal{A}|)}{IL_\beta(C_2, |\mathcal{A}|)} &= -\log p_0 < 1/D \end{aligned}$$

□

## 5 Experimental Results

In this section, we report results obtained by applying our technique to compare information leakage of two programs. We begin by reexamining PROG P4 using our Algorithms. Consider four different parameter values of  $L$ :  $L = |\mathcal{A}|/c$ ,  $L = c \log |\mathcal{A}|$ ,  $L = c\sqrt{|\mathcal{A}|}$ ,  $L = c$  where  $c > 2$  is certain constant. Then,

$$\begin{cases} L = |\mathcal{A}|/c, IL_\infty(P4, |\mathcal{A}|) = \log\left[\frac{c}{c-1}\right] \\ L = c \log |\mathcal{A}|, IL_\infty(P4, |\mathcal{A}|) = -\log\left[1 - \frac{\log |\mathcal{A}|}{|\mathcal{A}|}\right] \approx \frac{\log |\mathcal{A}|}{|\mathcal{A}|} \\ L = c\sqrt{|\mathcal{A}|}, IL_\infty(P4, |\mathcal{A}|) = -\log\left[1 - \frac{c\sqrt{|\mathcal{A}|}}{|\mathcal{A}|}\right] \approx \frac{c}{\sqrt{|\mathcal{A}|}} \\ L = c, IL_\infty(P4, |\mathcal{A}|) = -\log\left[1 - \frac{c}{|\mathcal{A}|}\right] \approx \frac{c}{|\mathcal{A}|} \end{cases}$$

According to definition 4.6, for PROG P4, when  $L = |\mathcal{A}|/c$ , the leakage level is  $\Theta(1)$ ; when  $L = c \log |\mathcal{A}|$ , the leakage level is  $\Theta(\log |\mathcal{A}|/|\mathcal{A}|)$ ; when  $L = c\sqrt{|\mathcal{A}|}$ , the leakage level is  $\Theta(1/\sqrt{|\mathcal{A}|})$ ; when  $L = c$  the leakage level is  $\Theta(1/|\mathcal{A}|)$ . PROG P4 leaks more information as  $L$  decreases. The result matches the intuition of program flow leakage. Indeed as  $|\mathcal{A}| \rightarrow \infty$  then  $L = \frac{|\mathcal{A}|}{c}$  leaks non-zero information (e.g., when  $c = 2$  the program leaks one bit of information); while for all other values of  $c$  considered above the program leaks almost no information.

Let us now consider program P5 (see below):  $A$  is the high input and  $1 < L \in N^+$  is an integer parameter.

### PROG P5

$$O \equiv A \mod L$$

For any value of  $1 < L \in N^+$ ,  $\|\mathbf{p}_{|\mathcal{A}|}\|_\infty = \frac{[|\mathcal{A}|/L]}{|\mathcal{A}|} \rightarrow \frac{1}{L}$  as  $|\mathcal{A}| \rightarrow \infty$ , so  $IL_\infty(P5, |\mathcal{A}|)$  is finite for all  $|\mathcal{A}|$ . Thus P5 with any finite  $L$  has leakage level  $\Theta(1)$ , which indicates that P5 is on the same security level as P4 with  $L = |\mathcal{A}|/c$ .

Let us now consider another program P6 (see below):  $A$  is an integer with  $k$  bits ( $|\mathcal{A}| = 2^k$ ), and  $0 \leq L \leq k$  is an integer parameter.

### PROG P6

**if**  $A$  consists of  $L$  bits of 1 and  $k - L$  bits of 0 **then**

$$O = 1$$

**else**

$$O = 0$$

**end if**

Consider different values of  $L$ :  $L = 0, 1, 2, 3, \dots$  Then

$$IL_\infty(P6, 2^k) = -\log\left[1 - \frac{\binom{k}{L}}{2^k}\right] \approx \frac{\binom{k}{L}}{2^k}$$

Because  $\binom{k}{L}/\binom{k}{L+1} \rightarrow 0$  as  $k \rightarrow \infty$ , the leakage of PROG P6 increases as  $L$  increases. Actually, the leakage level of P6 is  $\Theta(k^L/2^k)$ . PROG P6 with  $L = 0$  has the same leakage level as

PROG P4 with  $L = c$ ; PROG P6 with  $L = 1$  has the same leakage level as PROG P4 with  $L = c \log |\mathcal{A}|$ .

We have admit with regret that Algorithm 4.4 still unable to distinguish all FOPs, take the following program for example, where  $A$  is the high input with  $k$ -bits and  $L \in N$  is an integer parameter.

**PROG P7**

```

if  $\log |\mathcal{A}| = k$  is even then
   $O \equiv A \pmod{2}$ 
else
   $O = 1_{\{A=L\}}$ 
end if

```

PROG P7 has leakage level  $\Theta(1)$  when  $\log |\mathcal{A}|$  is even, but has leakage level  $\Theta(1/|\mathcal{A}|)$  when  $\log |\mathcal{A}|$  is odd. When comparing P7 ( $L = 1$ ) with P4 ( $L = c \log |\mathcal{A}|$ ), we have  $f_\infty = \infty$  but  $g_\infty = 0$ . It is not applicable to determine a constant leakage level of P7 since it switches between high and low leakage constantly.

## 6 Summary

In this paper we point out important drawbacks in past approaches to information-theoretic measures for quantifying program leakage. We show using examples that some of the metrics proposed by past work may not only be counter-intuitive but also conflict with each other. We have presented a novel conflict-free approach to compare information leakage in two programs and show that it may be impossible to derive a more fine-grained comparison using Renyi-entropy based leakage measures. Using several examples we show that the proposed approach vastly outperforms past approaches in matching popular consensus on program information leakage.

## References

- [1] M. Backes, B. Kopf, and A. Rybalchenko, “Automatic discovery and quantification of information leaks,” in *IEEE Symposium on Security and Privacy*, pp. 141–153, 2009.
- [2] J. Wittbold and D. Johnson, “Information flow in nondeterministic systems,” in *IEEE Symposium on Security and Privacy*, pp. 144–161, 1990.
- [3] M. Clarkson, A. Myers, and F. Schneider, “Belief in information flow,” in *18th IEEE Computer Security Foundations Workshop*, pp. 31–45, 2005.
- [4] A. Sabelfeld and A. C. Myers, “Language-based information flow security,” *IEEE Journal on Selected Areas of Communication*, 2003.
- [5] R. Giacobazzi and I. Mastroeni, “Abstract non-interference: Parameterizing non-interference by abstract interpretation,” in *ACM Symposium on Principles of Programming Languages*, 2004.

- [6] S. Hamadou, V. Sassone, and C. Palamidessi, “Reconciling Belief and Vulnerability in Information Flow,” in *IEEE Symposium on Security and Privacy*, pp. 79–92, 2010.
- [7] G. Smith, “On the foundations of quantitative information flow,” *Foundations of Software Science and Computational Structures*, pp. 288–302, 2009.
- [8] J. Pliam, “On the Incomparability of Entropy and Marginal Guesswork in Brute-Force Attacks,” *Progress in Cryptology—INDOCRYPT 2000*, pp. 113–123, 2000.
- [9] A. Teixeira, A. Souto, A. Matos, and L. Antunes, “Entropy measures vs. algorithmic information,” *ArXiv e-prints*, Jan. 2009.
- [10] C. Cachin, *Entropy measures and unconditional security in cryptography*. Zürich, 1997.
- [11] J. L. Massey, “Guessing and entropy,” in *IEEE International Symposium on Information Theory*, p. 204, 1994.
- [12] A. RRNYI, “On measures of entropy and information,” *Fourth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 547–561, 1961.
- [13] T. Cover and J. Thomas, *Elements of information theory*. John Wiley and Sons, 2006.
- [14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, 3 ed., 2009.